IN THE CLAIMS:

Please amend claims 1, 9, 10, 12, 24, 27, 28, 34, and 35 and cancel claims 2-8, 25 and 26 without prejudice or disclaimer as follows:

1. (Currently amended) A data network node switching Protocol Data Units (PDUs) between a plurality of data ports, each PDU having a header storing header information and a payload storing at least a portion of data to be conveyed, the data network node comprising:

a. a processor switching PDUs; and

b. a shared memory buffer ~~including~~ comprising:

i. a reserved temporary memory storage portion for holding, while pending a preliminary inspection of the header information prior to queuing, PDUs received via an input port of the plurality of data ports; and

ii. a PDU queuing memory storage portion for holding PDUs pending processing in determining at least an output port from the plurality of data ports to switch the PDU to, the PDU queuing memory storage portion further comprising a Classes-of-Service (CoS) PDU queuing portion including a plurality of reserved CoS processing queues, each CoS processing queue holding, while pending processing, PDUs associated with a one of a plurality of classes-of-service supported at the data network node in providing support for Quality-of-Service guarantees ensuring the availability of minimum memory storage resources for data flows at the data network node,

whereby the arrangement reduces PDU discard instances at the data network node by enabling the discrimination of PDUs associated with well-behaved data flows from PDUs associated with misbehaving data flows prior to queuing thereof.

2-8. Cancelled

9. (Currently amended) A data network node as claimed in claim ~~8~~ 1, wherein a CoS processing queue is designated for storing while processing, PDUs without a specified class-of-service (best effort data traffic PDUs).

10. (Currently amended) A data network node as claimed in claim ~~8~~ 1, wherein at least one CoS processing queue has an adjustable memory size.

11. (Original) A data network node as claimed in claim 10, wherein the memory size of at least one CoS processing queue is modified via one of: a management console and a higher level protocol enforcing Quality-of-Service (QoS) guarantees.

12. (Currently amended) A data network node as claimed in claim ~~8~~ 1, wherein the data network node further comprises a plurality of CoS processing queue storage bits, each CoS processing queue storage bit being associated with a PDU stored in CoS processing queues.

13. (Original) A data network node as claimed in claim 1, wherein the PDU queuing memory storage portion further comprises a shared memory-pool portion holding, while pending processing, PDUs associated with data flows conveying PDUs at data rates above reserved data rates in providing QoS guarantees.

14. (Original) A data network node as claimed in claim 13, wherein the shared memory-pool portion further holds while pending processing PDUs without a specified class-of-service (best effort data traffic PDUs).

15. (Original) A data network node as claimed in claim 1, wherein the PDU queuing memory storage portion further comprises an input port PDU queuing portion including a plurality of reserved input port processing queues, each input port processing queue holding, while pending processing, PDUs associated with a one of a plurality of input data ports of the data network node providing additional storage for PDUs associated with data flows conveying PDUs via the input port whereby protection against blocking is provided for data flows conveyed via the input port from misbehaving data flows conveyed via other congested input ports.

16. (Original) A data network node as claimed in claim 15, wherein at least one input port processing queue has an adjustable memory size.

17. (Original) A data network node as claimed in claim 16, wherein the memory size of the at least one input port processing queue is modified via one of: a management console, a higher level protocol enforcing Quality-of-Service (QoS) guarantees and a higher level protocol enforcing flow control.

18. (Original) A data network node as claimed in claim 15, wherein each input port processing queue has an associated high watermark level for comparison against a level of occupancy of the input port flow control processing queue in effecting input port flow control.

19. (Original) A data network node as claimed in claim 18, wherein at least one high watermark level is adjustable.

20. (Original) A data network node as claimed in claim 19, wherein the value of the high watermark level is modified via a one of: a management console, a higher level protocol enforcing Quality-of-Service (QoS) guarantees and a higher level protocol enforcing flow control.

21. (Original) A data network node as claimed in claim 18, wherein each input port processing queue has an associated low watermark level for comparison against a level of occupancy of the input port processing queue in effecting input port flow control.

22. (Original) A data network node as claimed in claim 21, wherein at least one low watermark level is adjustable.

23. (Original) A data network node as claimed in claim 22, wherein the value of the low watermark level is modified via a one of: a management console, a higher level protocol enforcing Quality-of-Service (QoS) guarantees and a higher level protocol enforcing input port flow control.

24. (Currently amended) A method of processing Protocol Data Units (PDUs) at a data network node having a processor switching PDUs and a shared memory buffer including a PDU queuing memory storage portion further including a Class-of-Service (CoS) PDU memory storage portion, the CoS PDU memory storage portion further including a plurality of CoS processing queues associated with a group of classes-of-service supported at the data network node, the method comprising steps of:

a. receiving a PDU via an input data port of the data network node;

b. temporarily storing the received PDU in a reserved temporary memory storage portion of the shared memory buffer;

c. extracting header information from the stored PDU;

d. selectively queuing the PDU for processing in a corresponding CoS processing queue in accordance with a CoS specification held in the extracted header information if the CoS processing queue is not full;

e. switching the PDU;

f. transmitting the PDU via an output data port of the data network node; and

g. deallocating resources used by the transmitted PDU,

whereby the storage of the received PDU in the temporary memory storage portion prior to header inspection provides for a qualified determination to be made in discarding PDUs.

25. Cancelled

26. Cancelled

27. (Currently amended) A method as claimed in claim ~~26~~ 24, wherein queuing the PDU in the corresponding CoS processing queue, the method further includes a step of setting an associated CoS processing queue storage bit corresponding to each PDU queued therein.

28. (Currently amended) A method as claimed in claim ~~26~~ 24, wherein the PDU queuing memory storage portion further includes a shared memory-pool portion and the step of selectively queuing the PDU for processing, the method further comprises a step of: queuing the PDU in the shared memory-pool portion if the CoS processing queue corresponding to the PDU is full.

29. (Original) A method as claimed in claim 28, wherein the method further comprises a step of: primarily queuing PDUs without a specified class-of-service (best effort data traffic PDUs) in the shared memory-pool portion.

30. (Original) A method as claimed in claim 28, wherein the PDU queuing memory storage portion further includes an input port PDU queuing storage portion, the input port PDU queuing storage portion further includes a plurality of input port processing queues, each input port processing queue being associated with an input port of the data network node, and the step of selectively queuing the PDU for processing, the method further comprises a step of: queuing

the PDU in a input port processing queue corresponding to the input port on which the PDU was received if the shared memory-pool resources have been exhausted.

31. (Original) A method as claimed in claim 30, wherein the method further comprises the step of enforcing input port flow control.

32. (Original) A method as claimed in claim 31, wherein enforcing input port flow control each input port queue has an associated high watermark level against which a level of occupancy of the input port processing queue is compared in effecting flow control.

33. (Original) A method as claimed in claim 31, wherein enforcing input port flow control each input port flow control queue has an associated low watermark level against which a level of occupancy of the input port processing queue is compared against in clearing a congested state of the data network node.

34. (Currently amended) A method as claimed in claim 24, wherein deallocating resources used by the transmitted PDU, the method further comprises a step of: returning ~~the freed~~ memory space used by the transmitted PDU to a CoS processing queue if the PDU was queued for processing therein.

35. (Currently amended) A method as claimed in claim 34, wherein returning the ~~freed~~ memory space used by the transmitted PDU to a CoS processing queue the method further comprises a step of: determining whether a CoS processing queue storage bit associated with the transmitted PDU had been previously set.

36. (Original) A method as claimed in claim 30, wherein deallocating resources used by the transmitted PDU, the method further comprises a step of: returning the freed memory space to the corresponding input port processing queue if the level of occupancy thereof is not zero.

37. (Original) A method as claimed in claim 36, wherein deallocating memory resources used by the transmitted PDU, the method further comprises a step of: returning the freed memory space to the shared memory-pool portion if a corresponding flow control processing queue occupancy level is zero.